

Report Center

1. Introduction

The Report Center application is a tool for submitting reports by students and correcting them by assistants. Reports submitted by students will automatically appear when the responsible assistants login, and if they accept or refuse a report it will be displayed in the student's view.

2. Documentation User

2.1. Documentation Student

Timetable

View the date of your experiments. Experiments with an asterisk demand a report to be submitted after the experiment.

Submissions

Filter

Use this option to display only accepted, refused or pending reports, and to display or hide the details for each experiment.

Experiments

If you need to upload a report for an experiment, there is a "Upload" button. Click it to go to the upload page.

If you have to correct a report there is a text displayed indicating what you have to correct or you can download the commented report by clicking on "Download". After correcting the report you can upload it by clicking on "Upload".

If a report is accepted or refused, try clicking on "Download" to see if the assistant has submitted a commented version of your report.

For an explanation of the icons see "Caption" at the bottom of this page.

If you have submitted a wrong report send a mail to the assistant of this experiment using "Feedback". As soon as the assistant has deleted your wrong submission you can resubmit your report.

Upload

Upload File

Choose the file you want to upload and click on "Upload". You can upload multiple files by repeating this step. **Be sure to always click on "Upload" before clicking on "Save", otherwise the file will not be submitted! You see all files which will be submitted under "Uploaded Files".**

Uploaded Files

List of files which will be submitted. If you have submitted a wrong file click on "Remove" to reset the form and restart uploading files.

Comment

If you want you can write a comment for the assistant.

If you have uploaded all files for this experiment, click on "Save".

Status

View the status of your reports for all experiments. For an explanation of the icons see "Caption" at the bottom of this page.

Feedback

Use this form to send an email to assistants or administrators. Select a person and fill in the subject and message.

2.2 Documentation Assistant

Submissions

Filter

Use this option to display only pending or corrected reports, only reports for one experiment (if you have more than one experiment), change the sorting, and to display or hide the details for each experiment.

Experiments

If you need to correct a report it appears in this list. Click on the "Download" button to download the submitted report.

If you have corrected the report click on "Accept", "Revise" or "Refuse" to accept/revise/refuse the report.

If a student has submitted a wrong report or if you have accidentally refused (or accepted) a report, you can delete the last submission by clicking on "Remove".

For an explanation of the icons see "Caption" at the bottom of this page.

You see only submissions for experiments for which the administrator has declared you assistant. There can only be one assistant per experiment and only this person can accept or refuse reports!

Upload

You can upload the corrected/commented report, otherwise you can write the corrections in the comment field.

Upload File

Choose the file you want to upload and click on "Upload". You can upload multiple files by repeating this step. **Be sure to always click on "Upload" before clicking on "Accept"/"Revise"/"Refuse", otherwise the file will not be submitted! You see all files which will be submitted under "Uploaded Files".**

Uploaded Files

List of files which will be submitted. If you have submitted a wrong file click on "Remove" to reset the form and restart uploading files.

Comment

If you want you can write a comment for the student.

If you have uploaded all files you want to submit, click on "Accept"/"Revise"/"Refuse".

Status

List of all students showing the status for each experiment. For an explanation of the icons see "Caption" at the bottom of this page.

To view details for a student click on the name of the student.

Feedback

Use this form to send an email to students, other assistants or administrators. Select a person or a group (e.g. "All Students") and fill in the subject and message.

2.3 Documentation Administrator

Groups

Add Groups

Add groups of students for one or more experiments. **After adding new groups, be sure to edit the new groups for each experiment by clicking on "Modify" and specify the experiment date for each group, otherwise the experiment will not appear when students login! Check the status page and be sure that all students have a grey circle for all experiments they participate.**

Modify Groups

List of all experiments for which groups are specified with possibility to modify or remove groups. **Experiments will only appear when students login if their group has set up a date for the experiment!**

Modify Groups Form

Group Number and Size

Change this values if you need more groups or bigger groups. Click on "Refresh" to update view.

Groups are valid for the following experiments

Select all experiments for which this groups should be created (only available if adding new groups).

Group xy

Set up the groups by selecting the students from the drop-down-menus. **If you are modifying a group, be sure to enter the experiment date for each group otherwise it will not appear in the student's view!**

Experiments

Add Experiment

Choose a name, short name and assistant for the experiment and select if students must submit a report for this experiment.

There can be only one assistant for each experiment, and only this assistant can accept or refuse reports submitted by students for this experiment.

Experiments

List of all experiments with possibility to modify or remove an experiment.

Modify Experiment Form

Modify Experiment

Choose a name, short name and assistant for the experiment and select if students must submit a report for this experiment.

There can be only one assistant for each experiment, and only this assistant can accept or refuse reports submitted by students for this experiment.

Status

List of all students showing the status for each experiment. For an explanation of the icons see "Caption" at the bottom of this page.

To view details for a student click on the name of the student.

Feedback

Use this form to send an email to students, assistants or other administrators. Select a person or a group (e.g. "All Students") and fill in the subject and message.

2.4 Documentation Superadmin

Praktika

Add Praktika

To add a new praktikum, choose a shortname and name and click on add. For the shortname, please use format px_yy where x is the number of the praktikum and yy the year, e.g. "p3_04" for "Praktikum III Wintersemester 04" or "p2_05" for "Praktikum II Sommersemester 05". Do not use whitespaces!

After adding a praktikum, be sure to do the following:

- Open Silva with the indicated link (<http://praktikum.mat.ethz.ch/edit>).
- Give all students "Viewer"-rights for the folder "Student" and "Viewer +"-rights for all praktika which they participate (praktika folders inside folder "Student", e.g. "/login/student/px_xy").
- Give all assistants "Viewer"-rights for the folder "Assistant" and "Viewer +"-rights for all praktika which they participate (praktika folders inside folder "Assistant", e.g. "/login/assistant/px_xy").
- Give all administrators "Viewer"-rights for the folder "Administrator" and "Viewer +"-rights for all praktika which they are administrators for (praktika folders inside folder "Administrator", e.g. "/login/admin/px_xy").
- Setup downloads pages for the praktikum using Silva (inside praktika folders under "Student", "Assistant" and "Administrator").

The following can be done by all administrators too:

- Login as administrator.
- Setup experiments.
- Setup groups for each experiment.

Delete Praktika

To delete an existing praktikum, choose the praktikum from the drop-down-menu and select what you want to delete:

- Delete only reports:
All uploaded reports will be deleted, but nothing else. So you can still check e.g. when a certain student has submitted a report for a experiment, and when it has been accepted.
- Delete reports and submissions:
All uploaded reports and also all submission data will be deleted. Experiments and groups will not be deleted.
- Delete reports, submissions, groups and experiments:
Experiments and groups will also be deleted, but the site for the praktikum remains, so you can define new experiments and groups.
- Delete all:
The site for the praktikum will also be deleted so it will not appear on the login page anymore. If you have set up privileges for students and assistants, this will be deleted too.

To be sure that you do not delete a praktikum accidentally, you have to enter the code which appears on the page.

3. Documentation for developers

This application runs under Zope/Silva. All scripts are written in Python and use the Zope and Silva API's. Some scripts uses special classes and settings specific to the ETH system. Please see chapter 3.5 for more informations about this topic.

It is tested and works with the following versions:

- Zope 2.7.3
- Silva 1.1
- Python 2.3.4

3.1 Folder structure

/	contains welcome page and side-wide objects; no authentication necessary
/locals	images
/scripts	scripts which require no authentication
/login	from here on authentication is required
student	from here on "Viewer" privileges are required
p_skel	template for automatically created new praktika; from here on "Viewer +" privileges are required
p4_o4	silva files for praktikum p4_o4
...	...
assistant	from here on "Viewer" privileges are required
p_skel	template for automatically created new praktika; from here on "Viewer +" privileges are required
p4_o4	silva files for praktikum p4_o4
...	...
admin	from here on "Viewer" privileges are required
p_skel	template for automatically created new praktika; from here on "Viewer +" privileges are required
p4_o4	silva files for praktikum p4_o4
...	...
superadmin	from here on "Viewer" privileges are required
p_scripts	scripts available only to superadmin
p_uploads	temp folder for uploads
p_reports	folder for uploaded reports

p4_o4	folder containing uploaded reports for praktikum p4_o4
...	...
scripts	scripts which require authentication
sql	sql objects

3.2 Database structure

The application uses four MySQL tables with the following structures:

rc_experiments

```
CREATE TABLE `rc_experiments` (
  `e_id` int(11) NOT NULL auto_increment,
  `u_id` varchar(50) default NULL,
  `e_shortcode` varchar(10) default NULL,
  `e_name` varchar(255) default NULL,
  `report` char(1) default NULL,
  `praktikum` varchar(50) default NULL,
  PRIMARY KEY (`e_id`))
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- e_id unique experiment id
- u_id user id (links to user information in ETH LDAP)
- e_shortcode short name of experiment
- e_name name of experiment
- report 0 if no report is demanded, 1 if report is demanded
- praktikum short name of the praktikum to which this experiment belongs

rc_groups

```
CREATE TABLE `rc_groups` (
  `g_id` int(11) default NULL,
  `u_id` varchar(50) NOT NULL default 'o',
  `e_id` int(11) NOT NULL default 'o',
  PRIMARY KEY (`u_id`,`e_id`))
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- g_id unique group id
- u_id user id (links to user information in ETH LDAP)
- e_id experiment id (links to experiment information in table rc_experiments)

rc_help

```
CREATE TABLE `rc_help` (
  `id` int(11) NOT NULL auto_increment,
  `page` varchar(100) NOT NULL default "",
  `help` text, PRIMARY KEY (`id`))
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- id unique help id
- page page to which help belongs (e.g. "/login/assistant/submissions")

- help help text (“**” will be replaced by “” when outputted)

rc_schedule

```
CREATE TABLE `rc_schedule` (
  `g_id` int(11) NOT NULL default '0',
  `date` date default NULL,
  PRIMARY KEY (`g_id`))
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- g_id group id (links to group information in table rc_groups)
- date date of experiment

rc_submissions

```
CREATE TABLE `rc_submissions` (
  `s_id` int(11) NOT NULL auto_increment,
  `g_id` int(11) default NULL,
  `u_id` varchar(50) default NULL,
  `date` date default NULL,
  `status` char(1) default NULL,
  `comment` text,
  `index` tinyint(4) default NULL,
  PRIMARY KEY (`s_id`))
ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- s_id unique submission id
- g_id group id (links to group information in table rc_groups)
- u_id user id (links to user information in ETH LDAP)
- date date of submission
- status status of submission (0: first submission; 1: report has to be corrected; 2: resubmission of report; 3: report accepted; 4: report refused)
- comment comment for submission
- index index of submission (starting with 0, then following submission for the same group and experiment have incremental values for index)

3.3 How Zope and Silva privileges are used

This application uses the silva privileges system to secure access to the praktika data and to associate the different users with the right praktika.

Zope and Silva roles are used in the following manner:

Anonymous

Anonymous access is only allowed for the welcome page.

Authenticated

All pages residing under /login require at least basic authentication.

Viewer

“Viewer” rights are required to access the folders /login/student, /login/assistant, /login/admin and /login/superadmin.

Users which have “Viewer” rights to the folder /login/student are supposed to be students.
 Users which have “Viewer” rights to the folder /login/assistant are supposed to be assistants.
 Users which have “Viewer” rights to the folder /login/admin are supposed to be admins.

Beside being necessary for security reasons this is used by the system to determine the list of all students, assistants and admins.

Viewer +

“Viewer +” rights are required to access subfolders of /login/student, /login/assistant and /login/admin.

Users which have “Viewer +” rights to a specific praktikum folder under /login/student are supposed to be students which participate at that praktikum. If e.g. user xy has “Viewer +” rights for the folder /login/student/praktikum_xy he is supposed to participate at praktikum_xy as a student.

Users which have “Viewer +” rights to a specific praktikum folder under /login/assistant are supposed to be assistants of that praktikum.

Users which have “Viewer +” rights to a specific praktikum folder under /login/admin are supposed to be admins for that praktikum.

Beside being necessary for security reasons this is used by the system to determine the list of students, assistants and admins for a specific praktikum.

3.4 Code API

3.4.1 Python scripts

Python scripts in /scripts

list get_links()

Input -

Output list containing for each link a tuple with the following three elements:

1. int: 0 if link is not selected, 1 if link is selected
2. string: link title
3. string: link url

Used to get the links for the menu navigation.

string html_encode(string string)

Input string to be html encoded

Output html encoded string

Html encodes a string so that it can be used in html code (converts also \n to
).

Python scripts in /login/scripts

void delete_reports_for_group(int id)

Input group id

Output -

Deletes all submitted reports for the passed group.

list get_admins()

Input -

Output list containing user id's

Returns all admins for the current praktikum. Admins are defined as users which have "Viewer +" privileges for the admin folder of the current praktikum (/login/admin/praktikum_xy).

list get_assistants()

Input -

Output list containing user id's

Returns all assistants for the current praktikum. Assistants are defined as users which have "Viewer +" privileges for the assistant folder of the current praktikum (/login/assistant/praktikum_xy).

string get_email_with_id(string id)

Input user id

Output email address

Returns the email address for the user with the given user id.

list get_groups_for_experiment(int id)

Input experiment id

Output list containing for each group a dictionary with the following keys:

- id: group id
- members: list containing all user id's of this group
- date: date of the experiment for this group

Returns a list with all groups (including group members and experiment date) for a given experiment id.

list get_groups_for_experiments(list ids)

Input list of experiment ids

Output list containing group id's

Returns a list with all group id's for the given experiment id's.

list get_groups_for_user(string u_id)

Input user id

if no user id is passed, the user id of the current user is used

Output list containing group id's

Returns a list with all group id's which the given user is member of.

list get_links()

Input -

list get_links()

Output list containing for each link a tuple with the following three elements:

1. int: 0 if link is not selected, 1 if link is selected
2. string: link title
3. string: link url

Used to get the links for the menu navigation.

string get_praktikum()

Input -

Output short name of the current praktikum

Returns the current praktikum short name (as used in the url, e.g. "p4_0405").

string get_praktikum_fullname()

Input -

Output name of the current praktikum

Returns the current praktikum name (e.g. "Praktikum IV WSo4/05").

list get_schedule(string u_id)

Input user id
if no user id is passed, the user id of the current user is used

Output list containing for each experiment a dictionary containing the following keys:

- g_id: group id
- u_id: user id
- e_id: experiment id
- date: date of the experiment
- e_name: name of the experiment
- e_shortname: short name of the experiment
- report: int (0 if no report is demanded, 1 if a report is demanded)

Returns the schedule for the user with the given user id.

list get_students()

Input -

Output list containing user id's

Returns all students for the current praktikum. Students are defined as users which have "Viewer +" privileges for the student folder of the current praktikum (/login/student/praktikum_xy). If there are no students with "Viewer +" privileges it returns all users for the current praktikum by fetching it from the rc_groups table (this can be useful for finished praktikums).

string get_user()

Input -

Output user id of the current user

Returns the user id of the current user.

string get_user_with_id(string id)

Input user id

Output full name

Returns the full name of the user with the given user id.

string get_username()

Input -

Output full name of the current user

Returns the full name of the current user.

string get_users_for_group(int id)

Input group id

Output string with full names of users

Returns a string with the full names of all users of the group with the given group id (e.g. "Stefan Muster, Peter Müller, Johann Meier")

string html_encode(string string)

Input string to be html encoded

Output html encoded string

Html encodes a string so that it can be used in html code (converts also \n to
).

void mail(string mail_from, string mail_to, string subject, string body)Input mail_from: sender of the mail
mail_to: receiver of the mail
subject: subject of the mail
body: mail body

Output -

Sends a mail using the local mail host.

Python scripts in /login/superadmin/p_scripts**void delete_reports_for_praktikum(string praktikum)**

Input praktikum short name (e.g. "p4_0405")

Output -

Deletes all submitted reports for the given praktikum.

int reset_praktikum(string praktikum, int type)

Input praktikum: praktikum short name (e.g. "p4_0405")
 type: 0 delete only reports
 1 delete reports and submissions information
 2 delete reports, submissions information and groups/experiments
 3 delete whole praktikum including silva sites; deletes folders:
 /login/student/praktikumxy
 /login/assistant/praktikumxy
 /login/admin/praktikumxy)

Output 0 if illegal type was passed, otherwise 1

Resets or deletes a praktikum (by deleting only submitted reports; or also submissions, groups etc.; or whole praktikum including silva sites).

3.4.2 SQL objects**SQL objects in /login/sql****add_experiment**

Input name name of the experiment
 shortname short name of the experiment
 assistant user id of the responsible assistant
 report 0 if no report is demanded, 1 if report is demanded

Output -

Adds an experiment to the current praktikum.

add_group

Input g_id group id
 u_id user id
 e_id experiment id

Output -

Adds a user-group relation for a given experiment.

add_schedule_for_group

Input g_id group id
 date date of the experiment

Output -

Sets the experiment date for a given group.

add_submission

Input g_id group id
 u_id user id
 status status of the submission
 index index of the submission
 comment comment of the submission

Output -

add_submission

Adds a submission for a given group. For details about status values and index see chapter 3.2.

delete_experiment

Input id experiment id

Output -

Deletes the experiment with the given id.

delete_experiments_for_praktikum

Input praktikum praktikum short name (e.g. "p4_o4o5")

Output -

Deletes all experiments for the given praktikum.

delete_group

Input g_id group id

Output -

Deletes the group with the given group id.

delete_groups_for_experiment

Input id experiment id

Output -

Deletes all groups for the given experiment.

delete_schedule_for_group

Input g_id group id

Output -

Deletes the schedule for the given group.

delete_submission

Input id submission id

Output -

Deletes the submission with the given submission id.

delete_submissions

Input where where-clause to define which submission should be deleted, e.g.:
 "g_id=55 OR g_id=56"

Output -

Deletes the submissions which correspond to the given where-clause.

get_assistant_for_experiment

Output u_id user id

get_assistant_for_experiment

Output u_id user id

Returns the user id of the assistant which is responsible for the given experiment.

get_experiment

Input id experiment id

Output e_id experiment id

 u_id user id

 e_shortcode experiment short name

 e_name experiment name

 report 0 if no report is demanded, 1 if report is demanded

 praktikum praktikum to which the experiment belongs

Returns information for experiment with given id.

get_experiment_for_group

Input id group id

Output e_id experiment id

Returns the experiment id for the group with the given id.

get_experiments

Input -

Output e_id experiment id

 u_id user id

 e_shortcode experiment short name

 e_name experiment name

 report 0 if no report is demanded, 1 if report is demanded

 praktikum praktikum to which the experiment belongs

Returns information for all experiments of the current praktikum.

get_experiments_for_assistant

Input id user id

Output e_id experiment id

 u_id user id

 e_shortcode experiment short name

 e_name experiment name

 report 0 if no report is demanded, 1 if report is demanded

 praktikum praktikum to which the experiment belongs

Returns information for all experiments which the assistant with the given id is responsible for.

get_experiments_for_praktikum

Input praktikum praktikum short name (e.g. "p4_0405")

get_experiments_for_praktikum

Output	e_id	experiment id
	u_id	user id
	e_shortname	experiment short name
	e_name	experiment name
	report	0 if no report is demanded, 1 if report is demanded
	praktikum	praktikum to which the experiment belongs

Returns information for all experiments of the given praktikum.

get_group

Input	u_id	user id
	e_id	experiment id
Output	g_id	group id

Returns the group id for a given user and experiment.

get_group_members_for_experiment

Input	id	experiment id
Output	g_id	group id
	u_id	user id

Returns all group id's and user id's for a given experiment.

get_groups_for_experiment

Input	id	experiment id
Output	g_id	group id

Returns all group id's for a given experiment

get_groups_for_user

Input	u_id	user id
Output	g_id	group id

Returns all group id's for a given user.

get_experiment_for_group

Input	id	group id
Output	e_id	experiment id

Returns the experiment id for the group with the given id.

get_new_group_id

Input	-	
Output	MAX(g id)+1	group id

Returns a free group id to be used for a newly created group.

get_schedule

Input where where-clause to define which schedule should be returned, e.g.:
 “rc_schedule.g_id=55 OR rc_schedule.g_id=56”

Output g_id group id
 u_id user id
 e_id experiment id
 date experiment date
 e_name experiment name
 e_shortname experiment shortname
 report 0 if no report is demanded, 1 if report is demanded

Returns the complete schedule (for all experiments, including experiments information) for the groups defined in the where-clause.

get_schedule_for_group

Input group group id
Output g_id group id
 date experiment date

Returns the schedule for the specified group.

get_submissions_for_group

Input group group id
Output s_id submission id
 g_id group id
 u_id user id
 date submission date
 status submission status
 comment submission comment
 index submission index

Returns all submissions for the given group. For details about status values and index see chapter 3.2.

get_users_for_group

Input id group id
Output u_id user id

Returns all user id's for a given group.

get_users_for_praktikum

Input praktikum praktikum short name (e.g. p3_0405)
Output u_id user id

Returns all user id's for a given praktikum by retrieving it from the rc_groups table. This method is used if there are no students with “Viewer +” privileges for this praktikum (which can be the case if it is a finished praktikum).

update_experiment		
Input	e_id	experiment id
	u_id	user id of the responsible assistant
	e_name	name of the experiment
	e_shortcode	short name of the experiment
	report	0 if no report is demanded, 1 if report is demanded
Output	-	
Updates experiment information for the experiment with the given id.		

3.4.3 Silva code sources

Silva code sources in /

get_help
Shows help for the current page.

Silva code sources in /login

get_caption
Shows a caption describing the status icons (used on the status and submissions pages).

get_downloads
Shows the downloads for the passed submission.

get_feedback_form
Shows the feedback form (used on the feedback pages).

get_status_all
Shows the status of all students (used on the status pages).

get_submissions_for_user
Shows all submissions for the passed user.

get_status_all
Shows the status of all students (used on the status pages).

get_upload_confirmation
Finishes the upload, performs the submission and shows a confirmation of the report submission.

get_upload_form
Shows the upload form to submit a report.

process_feedback_form
Sends the feedback from the feedback form (normally on the same page than get_feedback_form).

Silva code sources in /login/student**get_filter**

Shows the filter form to filter the result on the submissions page.

get_status

Shows the status of the current user.

get_submissions

Shows the submissions for the current user.

get_timetable

Shows the timetable for the current user.

Silva code sources in /login/assistant**get_filter**

Shows the filter form to filter the result on the submissions page.

get_submissions

Shows the submissions for the current user.

Silva code sources in /login/admin**get_experiments**

Shows the experiments for the current praktikum.

get_experiments_form

Shows the form to add new experiments.

get_groups

Shows the existing groups.

get_groups_form

Shows the form to add new groups.

get_modify_experiment_form

Shows the form to modify existing experiments.

get_modify_groups_form

Show the form to add or modify groups.

process_experiments_form

Processes the experiment forms (add/modify/remove experiments).

process_groups_form

Processes the group forms (add/modify/remove groups).

Silva code sources in /login/superadmin**get_add_praktikum_form**

Shows and processes the form to add new praktika.

get_delete_praktikum_form

Shows and processes the form to delete existing praktika.

3.5 Information about portability of code to non ETH systems

This application uses the ETH LDAP functions to get informations about users. Silva users are not stored locally nor are they created using Zope. Instead users are stored on a LDAP server and authentication is done via LDAP too. Therefore some of the scripts in this application are specific to the ETH system and may not work on other systems.

Following a list of scripts which uses ETH specific functionality:

- get_email_with_id()
- get_students() -> SOBALD GRUPPEN-TOOL FUNKTIONIERT
- get_assistants() -> SOBALD GRUPPEN-TOOL FUNKTIONIERT
- get_admins() -> SOBALD GRUPPEN-TOOL FUNKTIONIERT